

Návrh jednotky na měření směru a rychlosti větru

Design of a unit for measuring wind direction and speed

Nikolas Rafaelis

Bakalářská práce

Vedoucí práce: Ing. Martin Sobek, Ph.D.

Ostrava, 2021

Abstrakt

V Bakalářské práci se zabýváme stručným shrnutím teorie a vysvětlením různých druhů měření rychlosti a směru větru. Také práce zahrnuje celkový návrh jednotky pro měření rychlosti a směru větru. Dále zajišťujeme různé ochrany před zkratem, odfiltrování zákmitu atd. Pro vyhodnocení využíváme mikrokontrolér ESP32, který zpracovává signály a zasílá je na nadřazený systém komunikačním protokolem MODBUS. Veškeré algoritmy jsou naprogramovány v jazyce C++.

Klíčová slova

Anemometr, ESP32, měření rychlosti větru, měření směru větru

Abstract

In this bachelor's thesis we are writing about a brief summary of the theory and explaining different types of measurements for the speed and direction of wind. This thesis also includes the whole design of the unit for measuring wind direction and speed. In the next part we provide different kinds of protection against the short circuits, vibration filtering, etc. We use microcontroller ESP32 for evaluating the results, which processes signals and then sends them on the superior system by using the communication protocol MODBUS. All algorithms are programmed in the C++ language.

Keywords

Anemometer, direction and speed of the wind, ESP32, measuring speed of the wind

Poděkování

Chtěl bych poděkovat vedoucímu mé práce, panu Ing. Martinu Sobkovi, Ph.D. za poskytnutí simulátoru směru a rychlosti větru pomocí různých spínačů. Také za skvělou komunikaci a poskytnutí odborných rad a názorů. Taktéž bych chtěl poděkovat mé rodině, a především přítelkyni, za trpělivost, podporu a péči.

Obsah

ÚVOD	10
1 TEORETICKÝ ROZBOR	11
1.1 VÍTR.....	11
2 MĚŘÍCÍ METODY RYCHLOSTI VĚTRU	12
2.1 ROZDĚLENÍ ANEMOMETRU.....	12
2.2 ROZDÍL MEZI DOTYKOVÝMI A BEZDOTYKOVÝMI.....	12
2.3 BEAUFORTOVA STUPNICE	13
2.4 PŘÍKLADY MĚŘÍCÍCH METODY.....	14
2.4.1 Mechanické anemometry	14
2.4.2 Tlakové anemometry	16
2.5 TEPELNÉ HMOTNOSTÍ PRŮTOKOMĚRY	18
2.6 BEZKONTAKTNÍ ANEMOMETRY.....	19
2.7 SMĚR VĚTRU	21
2.7.1 Příklady měření větru	22
3 NÁVRH JEDNOTKY	23
3.1 BLOKOVÉ SCHÉMA	23
3.2 SENZORY	24
3.2.1 Mechanický anemometr s rotačními miskami	24
3.3 ÚPRAVA SIGNÁLU	26
3.3.1 Úprava signálu pro mechanický anemometr s rotačními miskami	26
3.3.2 Úprava signálu pro mechanickou větrnou směrovku.....	28
3.4 MIKROKONTROLÉR ESP32	28
3.5 NAPÁJECÍ NAPĚTÍ + DC/DC MĚNIČ	29
3.6 ROZHRANÍ PRO NADŘAŽENÝ SYSTÉM	30
3.7 SOFTWARE	30
4 REALIZACE A OVĚŘENÍ.....	34
4.1 REALIZACE.....	34
4.2 OVĚŘENÍ.....	36
ZÁVĚR	39
LITERATURA	40
SEZNAM PŘÍLOH.....	41

Seznam použitých symbolů a zkratk

Zkratka	Význam
ADC	Analogově digitální převodník
DC-DC	Stejnoseměrný měnič
DPS	Deska plošných spojů
GPIO	Univerzální vstupní/výstupní pin
I2C	Sériová komunikační sběrnice
ISO/OSI	Referenční model
J	Jih
JJV	Jiho-jihovýchod
JJZ	Jiho-jihozápad
JV	Jihovýchod
JZ	Jihozápad
RS232	Sériový port
RS485	Standard sériové komunikace
S	Sever
SPI	Sériové periferní rozhraní
SRAM	Statická paměť
SSID	Identifikátor bezdrátové sítě
SSV	Severo-severovýchod
SSZ	Severo-severozápad
SV	Severovýchod
SZ	Severozápad

TCP/IP	Přenosový protokol síťové vrstvy
UART	Univerzální asynchronní přijímač – vysílač
V	Východ
VSV	Východo-severovýchod
VVJ	Východo-jihovýchod
Wi-Fi	Bezdrátová komunikace
Z	Západ
ZJZ	Západo-jihozápad
ZSZ	Západo-severozápad

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
C	F	Kondenzátor
f	Hz	Frekvence
h	m	Hloubka
I	A	Proud
n	-	Otáčky
r	m	Poloměr
R	Ω	Odpor
U	V	Napětí
v	m/s	Rychlost

Seznam ilustrací

Obrázek 1: Coriolisův jev[2]	11
Obrázek 2: rychlost otáčení Země[2]	11
Obrázek 3: mechanický anemometr	14
Obrázek 4: anemometr s rotačními lopatkami [5]	15
Obrázek 5: kyvadlový anemometr [6]	15
Obrázek 6: Pitotova trubice [7]	16
Obrázek 7: Prandtlova trubice [7]	17
Obrázek 8: Plovákové tlakoměry [7]	17
Obrázek 9: popis termoanemometrické sondy [8]	18
Obrázek 10: Akustický anemometr	19
Obrázek 11: Ultrazvukový anemometr	19
Obrázek 12: laser Doppler anemometr.....	20
Obrázek 13: směrová růžice [9].....	21
Obrázek 14: korouhev [9].....	22
Obrázek 15: větrná směrovka	22
Obrázek 16: blokové schéma	23
Obrázek 17: Anemometr s rotačními miskami	24
Obrázek 18: princip větrné směrovky	25
Obrázek 19: úprava signálu pro měření rychlosti proudění větru	26
Obrázek 20: úprava signálu pro mechanickou větrnou směrovku	28
Obrázek 21: vývody pinů ESP32 [12].....	28
Obrázek 22: DC-DC měnič	29
Obrázek 23: Komunikační rozhraní Modbus.....	30
Obrázek 24: Vyfrezovaná deska plošných spoju na cuprexitu	34
Obrázek 25: Osazená deska plošných spojů	34
Obrázek 26: Zapájena deska plošných spojů	35
Obrázek 27: Soustava senzorů s jednotkou	35
Obrázek 28: Simulátor rychlosti větru a směru	36
Obrázek 29: ošetřený signál proti zakmitávání.....	37
Obrázek 30: signál ovlivněný zakmitáváním	37

..

Seznam tabulek

Tabulka 1: rozdělení dle Beafortovy stupnice.....	13
Tabulka 2: naměřené úbytky napětí pro dané směry	26

Úvod

Proudící vzduch v atmosféře nebo na zemi, může způsobovat katastrofální problémy například hurikány, tsunami nebo změny teplotního klimatu, které zasahují do života lidí a mohou mnohé z nich o něj připravit. Proto je dobré o něm vědět základní parametry jako je směr a rychlost, které nám pomohou zamezit, předejít nebo upozornit na možné katastrofy.

Pro měření proudění větru využíváme anemometry, které pracují na různých fyzikálních jevech. Můžeme je vyjádřit pomocí tlaku, odrazu, tření nebo třeba laseru. Nejhojnější využití má v meteorologii, ale také ji můžeme najít v dopravních prostředcích nebo také v ochranných systémech zařízení, kde zabráňují poničení systému před větrem.

V této bakalářské práci si definujeme stručný výklad o proudícím vzduchu a popíšeme si výše zmíněné jevy a různé druhy měření rychlosti větru a směru. Na světě existují desítky způsobů metod měření, a proto si popíšeme jen nejzákladnější z nich.

Další částí bakalářské práce je návrh jednotky pro měření rychlosti větru a směru. Zabýváme se získáním převodu proudícího vzduchu na elektrický signál, který musí být správně vyfiltrován, omezen a nastaven pro vstup mikrokotroléru, který na základě algoritmu napsaném v jazyce C++ vyhodnotí výsledek a zašle jej na nadřazený systém.

Tato práce je vytvořena jako část projektu meteostanice, která už byla vytvořena vedoucím práce, avšak nebyla účinná, díky četným poruchám. Proto je projekt rozdělen do více částí, které budou různě rozmístěny pro efektivnější podmínky. Všechny části projektu mají společný bod a tím je nadřazený systém.

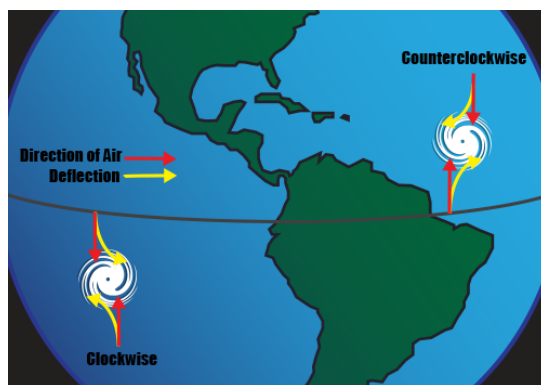
Problematika, náročnost, ale také informace, které díky této práci získáme, byly důvodem výběru tématu. Díky práci s programováním, návrhem plošné desky a obecně práci s elektrickým signálem, si prohloubíme znalosti v tomto směru a procvičíme si práci s elektrotechnickými součástkami.

1 Teoretický rozbor

1.1 Vítr

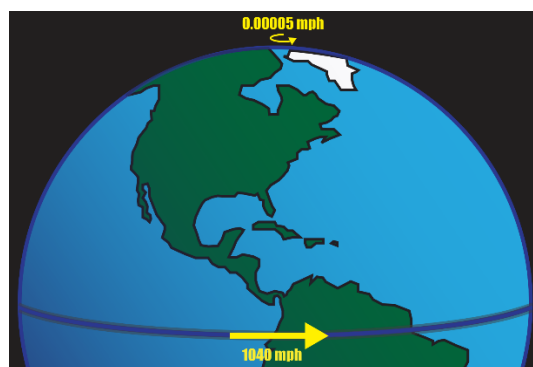
Vítr je pohyb vzduchu způsobený nerovnoměrným ohřevem Země sluncem. Je to fyzikální veličina, kterou sami o sobě necítíme a ani nevidíme, ale můžeme znát její sílu. Tento jev nám může sušit oblečení, dostat nás plachetnicí přes oceán, ale taky nám umí vytrhnout obrovský strom ze země. Může po světě přenášet teplo, vlhkost, znečišťující látky, prach atd. na velké vzdálenosti.

V atmosféře se větry mohou překládat, toto místo se nazývá **konvergenční zóna**. Ve většině případů silnější vítr fouká mezi směry východ – západ, než sever – jih. Tento fakt je způsobený rotací země neboli **Coriolisův jev**, který větrné směry otáčí proti směru hodinových ručiček na severní polokouli a na jižní polokouli se směr větru otáčí ve směru hodinových ručiček.[1]



Obrázek 1: Coriolisův jev[2]

Coriolisův jev ovlivňuje oceánské proudy, ale také i leteckou dopravu. Ve zkratce se dá o tom říct, že různé části země se pohybují různými rychlostmi. Zemi trvá 24 hodin, než se otočí. Jestliže budeme stát na severním nebo jižním pólu, Země se otočí asi o 0,00000805 km/h. O situaci na rovníku se totéž říct nedá, protože zde cestujeme po celém obvodu planety, to znamená, že cestujeme 1674,4 km/h, a to jenom při stání na místě. [2]



Obrázek 2: rychlost otáčení Země[2]

2 Měřicí metody rychlosti větru

K měření rychlosti větru slouží nástroj jménem Anemometr, který můžeme přeložit do češtiny „větroměr“. Největší využití má v oblasti meteorologie, kde se využívá pro měření rychlosti a směru proudění větru. Dále například v letectví a ostatních dopravních prostředcích. Při měření tohoto požadavku zjišťujeme velikost proudícího vzduchu kolem karoserie. Užitečný je také pro ochranu větrné elektrárny, kdy se díky němu zastaví mechanismus, pokud má příliš vysoké otáčky, nebo pro zefektivnění točení vrtule do lepších poloh.

2.1 Rozdělení anemometru

- Mechanické
- Tlakové
- Tepelné
- Bezdotykové/Laserové

2.2 Rozdíl mezi dotykovými a bezdotykovými

Měření větru můžeme rozdělit na 2 typy, a to kontaktní s větrem, jako je například anemometr s rotačními miskami, s vrtulí, termoanemometrické sondy atd., nebo bezkontaktně, například s principem Laser Doppler. Kontaktní metody spočívají ve vložení měřiče do proudu vzduchu, tato metoda je užitečná, ale ne tolik přesná, jelikož na systém můžou působit odchylky ve vlastnosti materiálu a dalších nežádoucích jevů. Tuto odchylku zkreslení můžeme kompenzovat koeficientem přepočtu na správnou hodnotu. U bezkontaktních metod se přístroje přímo nevystavují měřicí veličině (proudu větru). Tyto technologie využívají většinou princip vln, které zasahují do proudění. Pro tento vyspělejší systém se využívá ne jenom sofistikované optické zařízení, ale i také požadovaný software.

2.3 Beaufortova stupnice

Tabulka 1: rozdělení dle Beaufortovy stupnice[3]

Stupeň	Vítr	m/s	Tlak větru (kg/m2)
0	Bezvětrí	0 - 0,2	0
1	Vánek	0,3 - 1,5	0 - 0,1
2	Větrík	1,6 - 3,3	0,2 - 0,6
3	Slabý vítr	3,4 - 5,4	0,7 - 1,8
4	Mírný vítr	5,5 - 7,9	1,9 - 3,9
5	Čerstvý vítr	9 - 10,7	4 - 7,2
6	Silný vítr	10,8 - 13,8	7,3 - 11,9
7	Mírný vichr	13,9 - 17,1	12 - 18,3
8	Čerstvý vichr	17,2 - 20,7	18,4 - 26,8
9	Silný vichr	20,8 - 24,4	26,9 - 37,3
10	Plný vichr	24,5 - 28,4	37,4 - 50,5
11	Vichřice	28,5 - 32,6	50,6 - 66,5
12-17	Orkán	> 32,7	> 66,6

2.4 Příklady Měřících metody

2.4.1 Mechanické anemometry

Zde hovoříme o historicky nejstarší využívané měřicí metodě větru. U většiny mechanických anemometru se síla větru převádí na mechanickou konstrukci, která musí být umístěna ve směru síly větru, aby rotovala nebo se odchylovala z ustálené hodnoty. Umístění měřícího mechanismu nám může způsobit chyby, které se dají eliminovat výpočtem s koeficientem chybnosti.

Anemometr s rotačními miskami

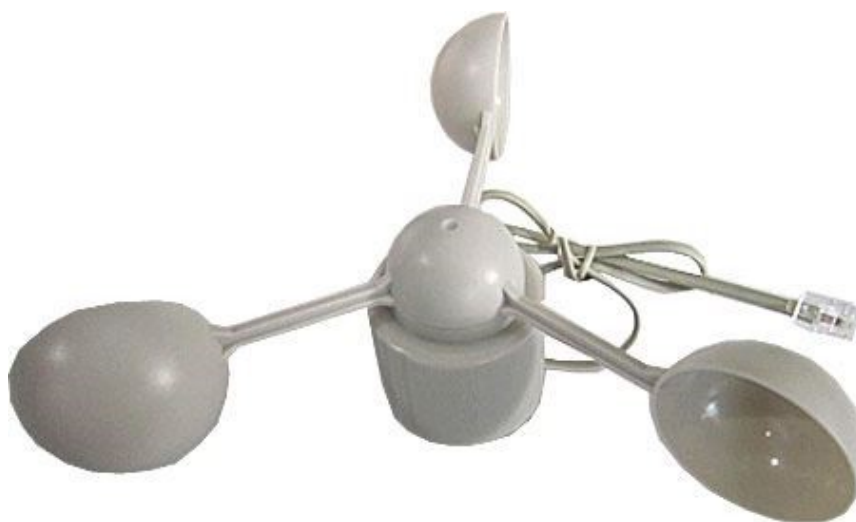
Zařízení je tvořené se čtyřmi nebo se třemi miskami na svislé ose. Princip je jednoduchý, jakmile se začne vzduch opírat do misek, vertikální hřídel se začne otáčet. Zde můžeme vidět značnou výhodu ve směru větru, kde nemusíme hlídat polaritu otáčení. Rozdílem mezi čtyř-miskovým a tří-miskovým anemometru je vliv úhlu proudu větru. U tří – miskovém anemometru získáváme dynamičtější záznam o větru.

$$v = r \cdot \omega \quad (1)$$

$$v = r \cdot 2 \cdot \pi \cdot f \quad (2)$$

$$v = r \cdot 2 \cdot \pi \cdot \frac{n}{t} \quad (3)$$

Kde n jsou otáčky, r je délka ramene a t je čas



Obrázek 3: mechanický anemometr

Anemometr s rotačními lopatkami

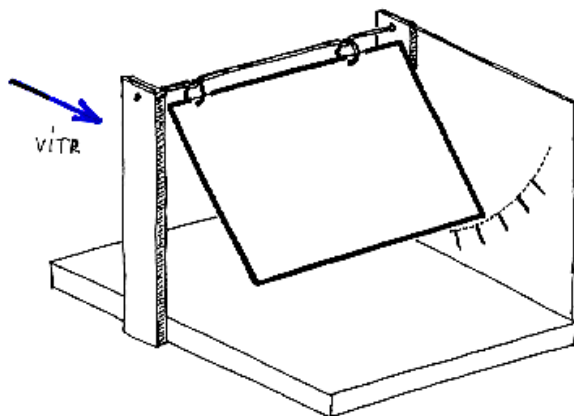
Je druhým nejběžnějším používaným mechanickým anemometrem. Konstrukce vrtulkového anemometru je obdobná jako v předchozím případě, jen s tím rozdílem, že je rotor vrtulky nasměřován horizontálně. U zmíněného anemometru je velká nevýhoda, že musí být vždycky orientován přesně proti větru. Proto je většinou na konstrukci přidáno křídlo, které směřuje směr větru správně. Tento způsob se označuje jako korouhev. Tato něčí metoda se dnes spíše používá jako měřicí přístroj. Anemometr je vybavený jak lopatkami, tak i elektronickým zařízením, které vyhodnocuje rychlost větru okamžitě, ale není tak přesný jako rotační anemometr s miskami.[4]



Obrázek 4: anemometr s rotačními lopatkami [5]

Kyvadlový anemometr

Jedná se o historický první anemometr. Princip je založený na nahnutí kyvadla, kdy, jakmile začne proudit vzduch na kyvadlo, se vytvoří odchylka, ze které je možné určit rychlost větru. Přepočet závisí na velikosti a taky na tvaru kyvadla. Tato metoda není zcela nejpřesnější.



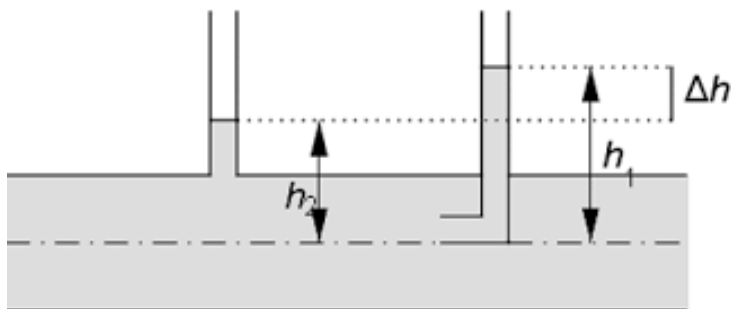
Obrázek 5: kyvadlový anemometr [6]

2.4.2 Tlakové anemometry

Rychlost větru se dá měřit pomocí tlaku vzduchu. Tlakový anemometr je většinou ve skleněné trubici, která je z jedné strany uzavřená a při proudění ideálního plynu nebo kapaliny v ideální hladké trubici mají všechny částice v průřezu stejný tlak a stejnou rychlost. Pro takovéto vlastnosti platí rovnice kontinuity. Tyto metody jdou realizovat pomocí Pitotovy nebo Prandtlovy trubice či jinými, kterými se v práci ale zabývat nebudeme.

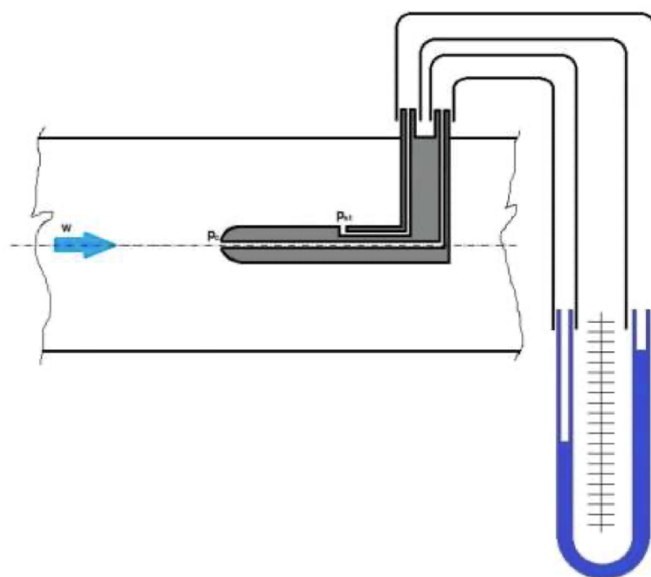
Pitotova trubice

Pitotova trubice je složena z tenkých trubiček vložených do proudu měřené látky, a to proti směru proudění. Konec jedné tenké trubičky je zahnutý a orientován proti směru proudění, zatímco druhý konec je kolmý směrem k proudící látce. Celkový tlak zjišťujeme pomocí dynamického tlaku p_d , který reaguje na vstup proudění, kdy se jeho kinetická energie přemění na tlakovou, protože vzduch se snaží udržet stejnou celkovou energii. Uvnitř druhého konce vzniká statický tlak p_s , který udává hodnotu hydrostatického tlaku uvnitř nepohybujícího se vzduchu. Obě trubice jsou spojeny do tvaru „U“ a v místě spojení je kapalina (většinou rtuť), která vytlačuje rozdíl tlaku. Z rozdílu tlaku získáváme výšku hladiny, pomocí které můžeme vypočítat rychlost v_p . Tuto metodu lze využít jenom pro měření plynů nebo velmi čistých kapalin z důvodu znečištění otvorů trubice. [7]



Obrázek 6: Pitotova trubice [7]

Prandtlova trubice

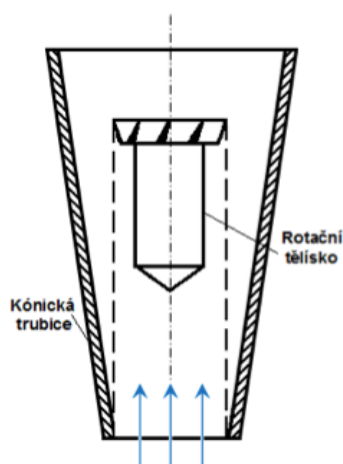


Obrázek 7: Prandtlova trubice [7]

Trubice je také založena na měření tlaku, ale pouze v jednom bodě. Zde se taky porovnává statický tlak s tlakem proudění. Odečet se provádí také z odečítání hladiny ve tvaru „U“. Jinak fungují na podobném principu jako trubice Pitotovy.

Plováčkové tlakoměry

Základem tohoto měření je svislá měřicí trubice, kterou prochází proudící médium, ve které je umístěný plováček. Pro úspěšné měření musí být hustota plováčku větší než hustota měřeného média. Proudící médium nadnáší plováček a jakmile se plováček ustálí, vztlaková síla se vyrovná jako gravitační, která na něj působí, v opačném směru. Ze zdvihu plováčku se určí velikost rychlosti proudění. Tvary plováčku se mohou lišit dle způsobu jejich použití. Poloha plováčku se odečítá buď přímo ze stupnice nebo fotoelektricky. Výhodou měření je snadná instalace a nízké náklady při měření nižších proudů. Nevýhodou je větší kalibrace. [4]



Obrázek 8: Plováčkové tlakoměry [7]

2.5 Tepelné hmotností průtokoměry

U tohoto měření vycházíme z tepelné závislosti mezi výměnou tepla zdroje a okolím. Tyto metody jsou užitečné pro měření malých průtoků. Nevýhodou je náročnější elektronika, a proto složitější kalibrace systému. Výhodou je okamžitý výsledek, který není tolik náchylný k vytváření chyb, jako třeba u měření tlakovým měřičem, kde může být únik média z potrubí nebo nedokonale hladká trubka atd.

Termoanemometrické sondy

Tato metoda závisí na elektrických odporech a jejich teplot. U měření se využívá tenký drátek (většinou se jedná o wolfram s průměrem v μm), který je nažhaven elektrickým proudem, zde musí být drátek nažhavený na teplotu větší než je teplota okolního vzduchu. Jakmile začne vzduch proudit kolem drátku, začne se ochlazovat, a tím mění své vlastnosti. Anemometr může být se stálým napětím proudu, nebo se stálou teplotou. Nevýhodou sondy je křehkost, ale naopak obrovskou výhodou je dynamická odezva a obrovská přesnost, proto je využíváme v potrubních systémech. [8]

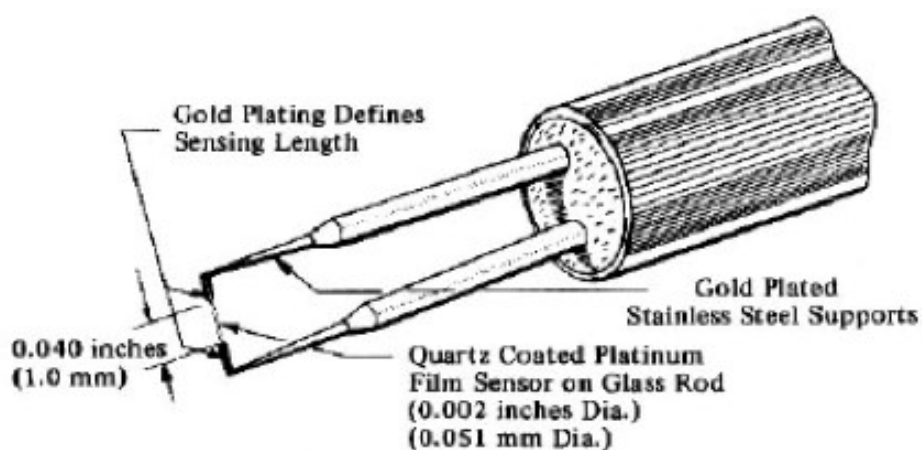


Figure 2: Cylindrical Hot Film Sensor and Support Needles- 0.002" Dia. (0.051 mm)

Obrázek 9: popis termoanemometrické sondy [8]

2.6 Bezkontaktní anemometry

Akustické anemometry

Princip činnosti spočívá v tom, že skrze prostředí jsou vysílány nadzvukové akustické vlny a výstupem je čas, za který vlna urazila svoji vzdálenost. Měření probíhá mezi vysílačem a přijímačem. Když využijeme více páru můžeme získat z měření jednodimensionální, dvoudimensionální nebo třídimensionální proudění. Vzdálenost mezi vysílačem a zesilovačem je většinou kolem 10–20 cm. Veliká výhoda tohoto měření je v dlouhodobém využití, jelikož se nám zde nepohybují žádné pohyblivé části, tudíž zde není nutná obrovská údržba. Nevýhoda spočívá v nepřesných výsledcích, které mohou být zkresleny například deštěm, který ovlivňuje šíření zvuku svojí hustotou. Veliké využití můžeme vidět v meteorologických stanicích, letectví apod. [14]



Obrázek 10: Akustický anemometr[13]

Ultrazvukové anemometry

Ultrazvukovým průtokoměr měří úbytek rychlosti šíření ultrazvukových vln. Princip spočívá ve dvou párech vysílacích a přijímacích jednotek, které jsou v rozestupu za sebou v měření potrubí, ale navzájem proti sobě. Tím pádem vysílač V1 se šíří ve směru a vysílač V2 proti směru. Výsledná hodnota je rozdíl mezi dobou přechodu. Ze které si můžeme odvodit rychlost proudění. Pro kvalitnější záznam rychlosti se využívá prodloužení přijímače a zesilovače.

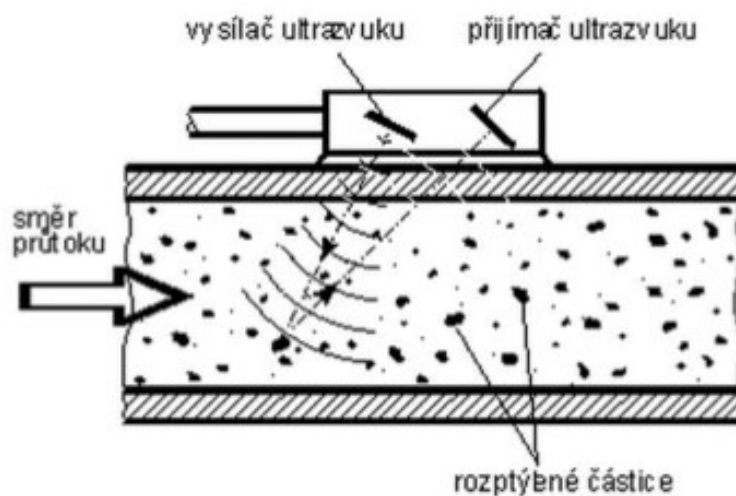


Obrázek 11: Ultrazvukový anemometr[13]

Laser Doppler anemometr

Celý princip je založený na Dopplerově jevu. Laserový paprsek prochází médiem, kde se paprsek v tekutině rozptyluje. Díky Dopplerova jevu zde dochází k porovnání frekvence odraženého paprsku a výsledného paprsku. Při porovnání lze získat výpočet rychlosti částice. Abychom mohli využívat tento princip musíme dobře uměle nadopovat tekutiny, tak aby měly malou spádovou rychlost a aby byly schopné kopírovat trajektorii proudu tekutiny. Dále abychom získali třídimenzionální vektor proudění musíme zařadit do děje tři zdroje světla o různých vlnových délkách. Abychom mohli zaregistrovat rozdíl frekvencí musí být zdroj světla monochromatický s velmi úzkou spektrální čarou. Vyhodnocení frekvenčních rozdílů lze s vysokou přesností zjistit z interferenčního obrazce z detektoru.

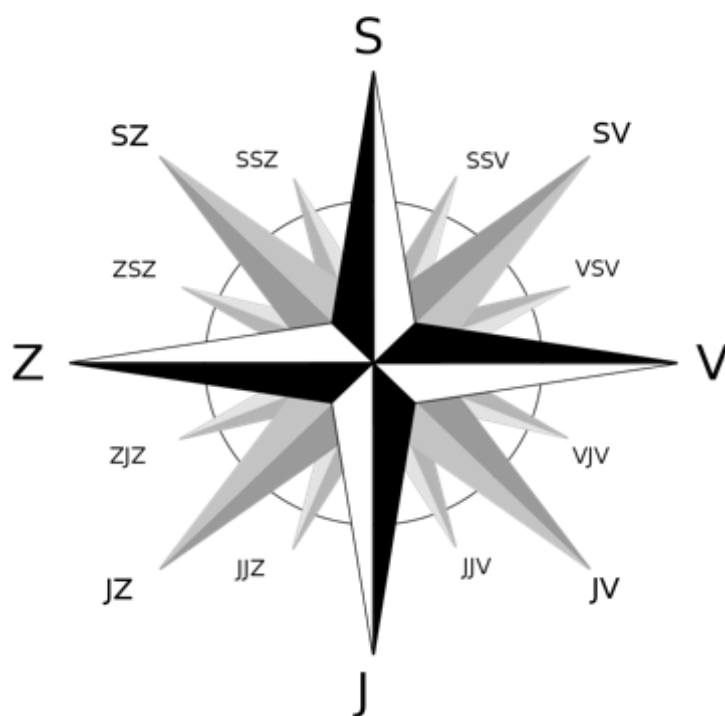
[12]



Obrázek 12: laser Doppler anemometr[7]

2.7 Směr větru

Směr větru je také důležitý faktor jak pro dopravu leteckou, tak vodní. Ze směru větru většinou odečítáme horizontální složku, a to většinou ve stupních. Například čistě severní vítr má hodnotu 0°. Hlavní strany jsou rozvržené po obvodě kruhu o posunu 90° jako je S, J, Z a V. Vedlejší strany jsou rozvržené po obvodě kruhu o posunu 45° jako jsou SZ, SV, JZ a JV. Pro směr větru využijeme ještě jemnější dělení, a to přidáním čárek, které budou po obvodě kruhu a posunu 22,5° jedná se například SSZ, JJV, VJV atd. Jemnějšímu dělení neboli dělení na 16 částí, se říká směrová růžice užitečná pro přesnější odhadování směru. U sensoru ultrazvukových nepotřebujeme jinou metodu k měření směru větru. [9]



Obrázek 13: směrová růžice [9]

2.7.1 Příklady měření větru

Korouhev

Pro zjišťování směru můžeme využít historicky oblíbenou korouhev, která funguje na mechanickém principu. Směr korouhve se natočí směrem, kde proudí vítr.



Obrázek 14: korouhev [9]

Větrná směrovka

Je to rozšířenější typ korouhve. Funguje na stejný princip, ale navíc má možnost zasílat svůj směr elektrickým signálem, většinou se jedná o úbytek napětí. Pro určení směru ze signálu potřebujeme navíc vyhodnocovací logiku, která rozeznává směr.

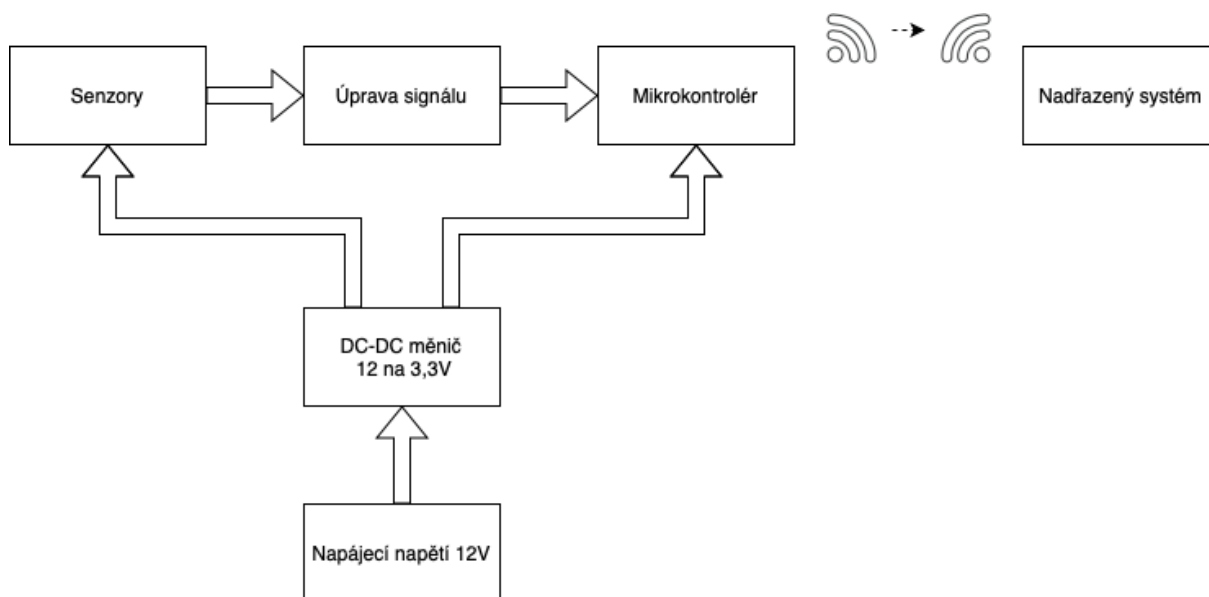


Obrázek 15: větrná směrovka

3 Návrh jednotky

Pro návrh jednotky pro měření směru a rychlosti větru jsme dostali zásadní informace pro konstrukci. Proudící vzduch bude měřený pomocí senzorů, které budou předávat signál mikrokontroléru, a ten signál následně zpracuje a pošle jej na nadřazený systém. Pro celkovou jednotku bude k dispozici napájecí napětí 12 V.

3.1 Blokové schéma



Obrázek 16: blokové schéma

3.2 Senzory

Pro získávání hodnot jsem využil mechanický anemometr s rotačními miskami a větrnou směrovku.

3.2.1 Mechanický anemometr s rotačními miskami

Vertikální vrtulka nám zaznamenává sepnutí magnetem kontakt. Pro jedno úplné otočení neboli o 360° se sepne dvakrát. Vypočtení koeficientu je důležitý pro přepočet z pulzů přerušení na rychlost v m/s. Vstupní napětí je přes $R = 10\text{ k}\Omega$ o hodnotě 3,3 V.

Můžeme vycházet z rovnice obvodové rychlosti, kde budeme počítat s časem za 1 s. Délka ramene je $r = 0,09\text{ m}$, otáčky jsou $n = 1$ otáčka.

$$v = r \cdot \omega \quad (4)$$

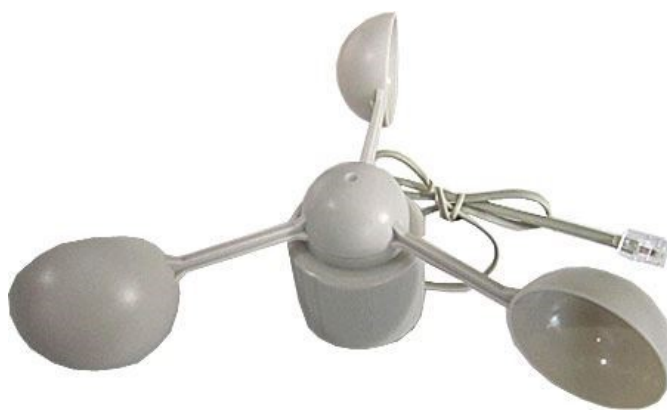
$$v = r \cdot 2 \cdot \pi \cdot f \quad (5)$$

$$v = r \cdot 2 \cdot \pi \cdot \frac{n}{t} \quad (6)$$

$$v = 0,09 \cdot 2 \cdot \pi \cdot 1 \quad (7)$$

$$v = 0,57\text{ m/s} \quad (8)$$

V našem případě vyšlo, že počet přerušení budeme násobit hodnotou 0,57, jenže to bychom počítali pro dvě otáčky za t, proto hodnotu $v = 0,57/2 = 0,28$. Výrobce udává 0,3 m/s, protože jsou zde započítané ztráty z misek anemometru, proto budeme počítat s hodnotou od výrobce.

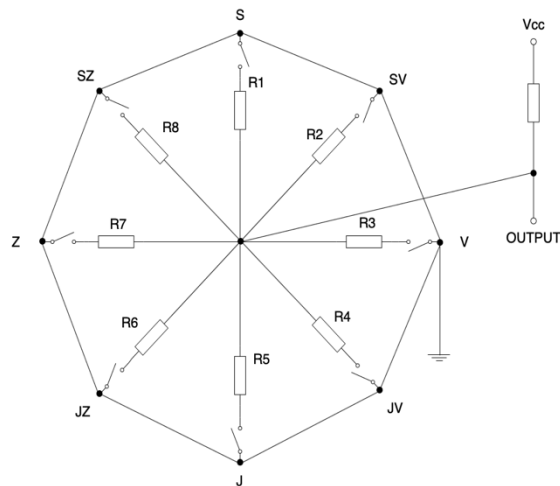


Obrázek 17: Anemometr s rotačními miskami

Větrná směrovka

Jedná se o složitější senzor pro měření směru, který se skládá ze osmi různě velkých odporů. Každý má svůj kontakt. V konstrukci je uložený magnet, který přitahuje různé kontakty podle směru natočení směrovky. Magnet nám může sepnout až dva kontakty, tím nám vytvoří 16 hodnot odporů. Pro získání informace o směru větru, využíváme dělič napětí pomocí přídavného odporu $R_{ucc} = 10\text{ k}\Omega$. Do senzoru prochází přes rezistor $U_{cc} = 5\text{ V}$ z kterého odečítáme úbytek U_{OUT} .

Příklad výpočtu úbytku U_{OUT} pro sever



Obrázek 18: princip větrné směrovky

Pro výpočet aplikujeme dělič napětí.

$$U_{OUT} = \frac{R_2}{R_1 + R_2} \cdot U_{Ucc} \quad (9)$$

$$U_{OUT} = \frac{33\,000}{10\,000 + 33\,000} \cdot 3,3 = 2,53\text{ V} \quad (10)$$

Tabulka 2: naměřené úbytky napětí pro dané směry

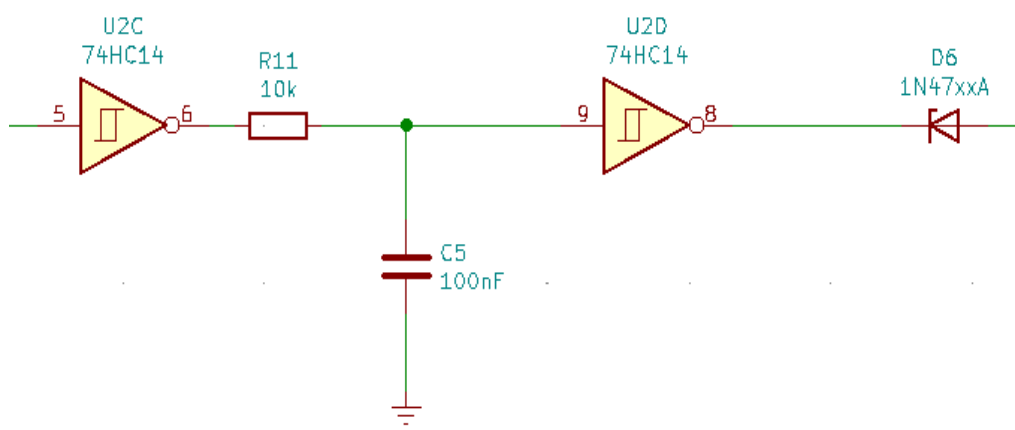
Směr	úhel (°)	odpor (Ω)	U _{mer} (V)	U _{vyp} (V)
S	0	3,3k	2,515	2,533
SSV	22,5	6,57k	1,3	1,308
SV	45	8,2	1,475	1,487
VSV	67,5	891	0,278	0,270
V	90	1k	0,3	0,300
VJV	112,5	688	0,21	0,212
JV	135	2,2k	0,588	0,595
JJV	157,5	1,41k	0,403	0,408
J	180	3,9k	0,918	0,926
JJZ	202,5	3,14k	0,78	0,789
JZ	225	16k	2,016	2,031
ZJZ	247,5	14,12k	1,918	1,932
Z	270	120k	3,041	3,046
ZSZ	292,5	42,12k	2,663	2,667
SZ	315	64,9k	2,856	2,859
SSZ	337,5	21,88	2,251	2,265

3.3 Úprava signálu

Abychom mohli získat plnohodnotný signál pro mikrokontrolér musíme ho upravit. Pro upravení signálu využijeme filtry, Schmittův spouštěč, sledovací komparátor, nebo třeba Schmittovu diodu.

3.3.1 Úprava signálu pro mechanický anemometr s rotačními miskami

Důležitým ošetřením signálu u anemometru s rotačními miskami je ošetření zákmitů a získávání pravoúhlého signálu pro dokonalejší výsledky.



Obrázek 19: úprava signálu pro měření rychlosti proudění větru

Pro filtrování zákmitu jsme využili dolní propust s hodnotami $R = 10\text{k}\Omega$ a $C = 100\text{ nF}$.

$$f_0 = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 10000 \cdot 0,0000001} 159,15\text{ Hz} \quad (11)$$

Po výpočtu mezní frekvence víme, že dolní propust bude propouštět maximálně 159 impulzů za sekundu.

Přepočet impulzů na m/s:

$$v = \text{impulzy} \cdot \text{koeficient} \quad (12)$$

$$v = 159 \cdot 0,3 \quad (13)$$

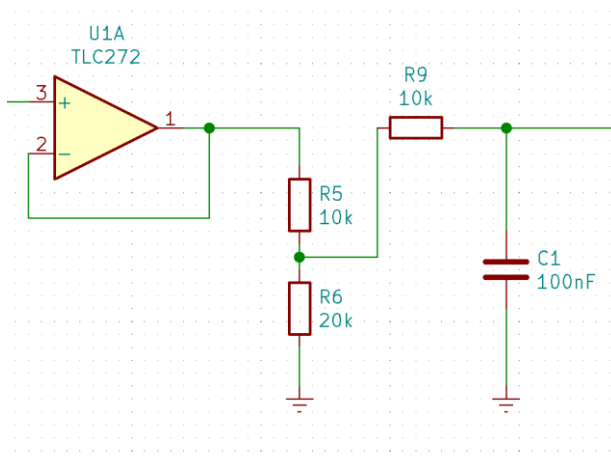
$$v = 47,7 \frac{\text{m}}{\text{s}} \rightarrow \frac{\text{km}}{\text{h}} = 47,7 \cdot 3,6 = 171,72 \text{ km/h} \quad (14)$$

Výsledek v je rychlost maximální pro měření větru pro tuto navrženou jednotku. Jelikož jednotka bude umístěna v okolí Fakulty elektrotechniky a informatiky, kde byla naměřena nejvyšší rychlost podle 119 km/h^[11]. Kdybychom chtěli uvažovat o umístění jednotky na Sněžku, kde byla naměřena nejvyšší rychlost větru 223 km/h^[11], musel by se vytvořit lepší filtr o větší mezní frekvenci.

Další úprava je vytvořena Schmittovým spouštěčem, kterou nám zastupuje integrovaný obvod 74HC14. Obvod obsahuje šest investujících komparátorů se Schmittovým spouštěčem. Funkčnost obvodu 74HC14 je v přeměňování vstupních signálů na ostré pravoúhlé signály bez zákmitu, co se týče napěťové úrovně přibližuje se vstupnímu napětí V_{cc} integrovaného obvodu. Pro ustálení úrovně napětí 3,3 V využíváme Zenerovu diodu o referenčním napětí 3,3 V. Výsledkem dvojího zapojení invertorů v sérii je získání zpátky neinvertovaného signálu.

3.3.2 Úprava signálu pro mechanickou větrnou směrůvku

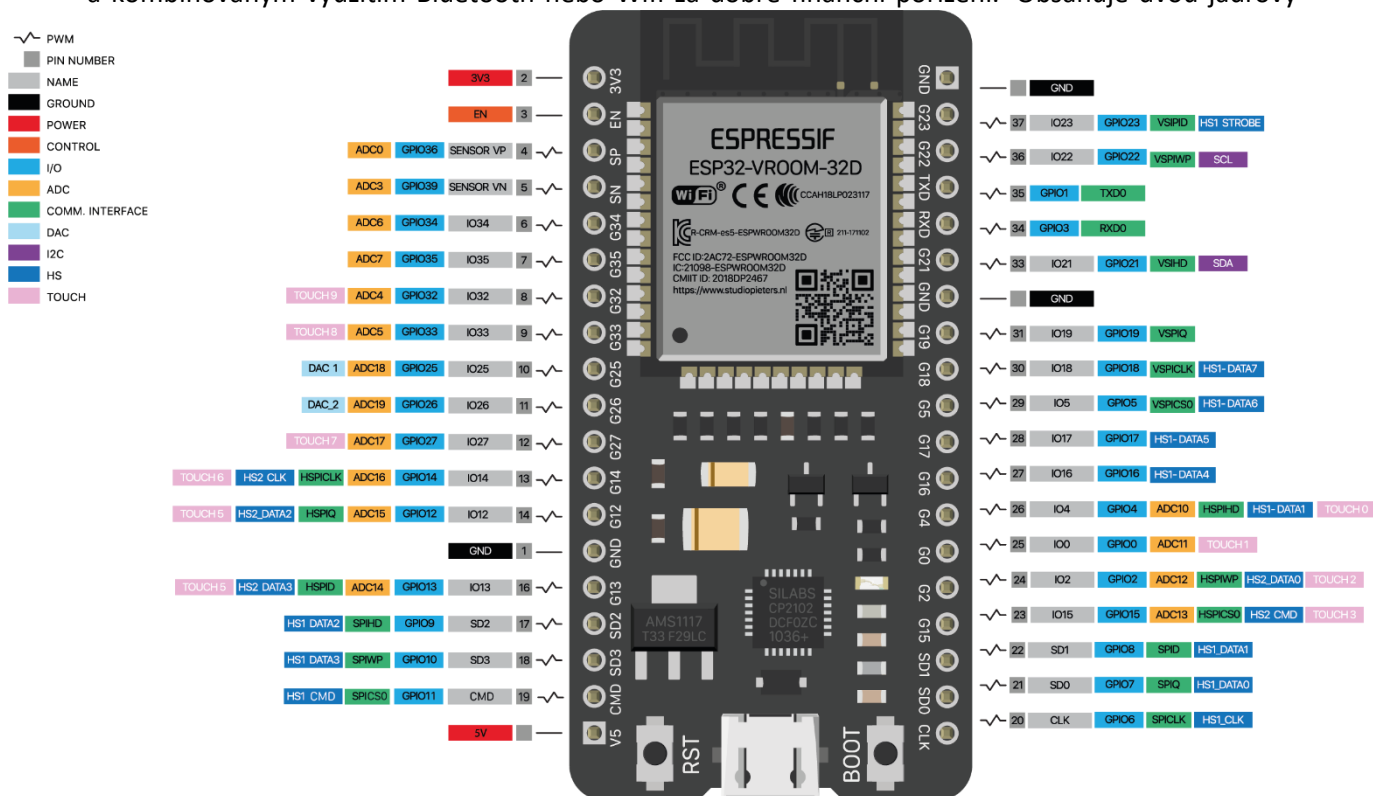
U větrné směrůvky jsme taky posilili vstupní signál pro mikrokontrolér napěťovým sledovačem pro menší impedanční zatížení s pomocí integrovaného obvodu TLC272. Pro snížení napětí z 5 V na 3,3 V jsme použili napěťový dělič. Jelikož v sensoru pracujeme s kontakty, musíme zde ošetřit zakmitávání pomocí stejného filtru, jako u sensoru měření rychlosti větru.



Obrázek 20: úprava signálu pro mechanickou větrnou směrůvku

3.4 Mikrokontrolér ESP32

ESP32 je vývojová deska od firmy Espressif. Výjimečná je svou nízko energetickou architekturou a kombinovaným využitím Bluetooth nebo Wifi za dobré finanční pořizování. Obsahuje dvou jádrový



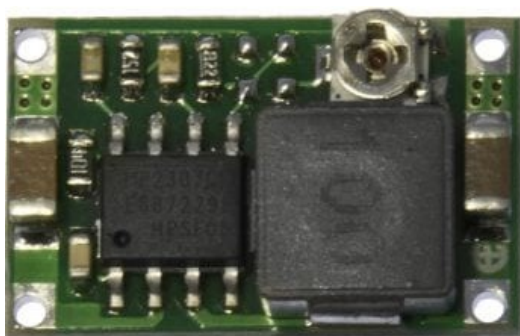
Obrázek 21: vývody pinů ESP32 [12]

procesor 32 bit s nastavovací frekvencí 80 MHz až 240 MHz. Pracuje se statickou pamětí SRAM 512 kB a integrovanou Flash pamětí 4 MB. Užitečná je také díky svým periferiím jako jsou, dotykové sensory, Hallův snímač, zesilovač s nízkým šumem, ethernet, UART, vysokorychlostní SPI a I2C. Napájecí napětí je 5 V a sám mikrokontrolér může vytvářet napětí o hodnotě 3,3 V. [12]

ESP32 se vyrábí pro různé počty pinů. My budeme využívat 38 pinovou desku, kde pracujeme s 34 GPIO. Ne všechny piny jsou určené pro plnohodnotné využití. Pro měření větru nebylo nutné zohlednit výběr pinu, protože jsme využívali pro získání větru přerušení, které je implementováno na každém GPIO. U měření větru jsme potřebovali ADC převodník, a proto jsme využili ADC0 u pinu GPIO36.

3.5 Napájecí napětí + DC/DC měnič

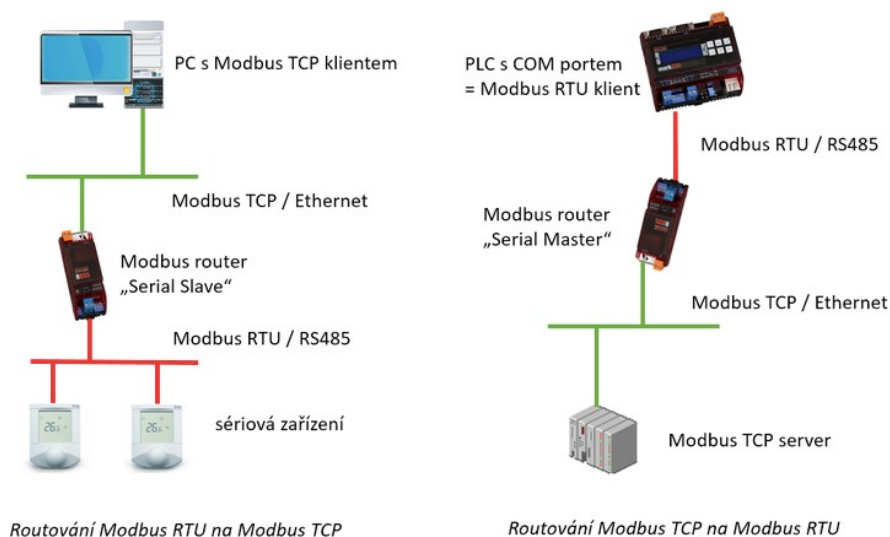
Pro snížení napětí z 12 V na 5 V, využijeme DC-DC měnič mini 360, který je na svoji prodejní cenu velmi efektivní. Můžeme s měničem vstupní napětí zredukovat od 4,75 V až po 23 V do 1 V – 17 V. Redukovanou hodnotu měníme pomocí hardwarového potenciometru. Velikost výstupního proudu může být až 1,8 A bez chladiče, s chladičem se můžeme dostat až na 3 A. Zakmitávání je vyřešené efektivními kondenzátory a tlumivkou.



Obrázek 22: DC-DC měnič

3.6 Rozhraní pro nadřazený systém

Modbus je komunikační protokol, který se používá v průmyslovém prostředí. Protokol je flexibilní a jednoduchý a lze ho přenášet po různých sběrnicích jako je RS232, RS485 nebo TCP/IP, kterou budeme využívat my. Topologie systému je založena mezi klientem a serverem. Server vyšle dotaz o data a klient mu odpoví. V ISO/OSI modelu pracuje na aplikační a fyzické vrstvě. [15]



Obrázek 23: Komunikační rozhraní Modbus[15]

3.7 Software

Program je vytvořený v praktickém integrovaném vývojovém prostředí PlatformIO IDE. Jedná se o prostředí se sadou různých nástrojů pro profesionální vývoj, které poskytují moderní a výkonné funkce pro zrychlení a zjednodušení vytvořených produktů. Zdrojový kód je napsaný v C++, ale můžeme jej psát i v jiném jazyce. Program pro tuto práci je rozdělen do více částí, ale přitom spolupracují paralelně. Nejzajímavější částí je využívání Wi-Fi protokolem TCP/IP pro přenos do nadřazeného systému.

Pro využití jednoduchosti a zrychlení systému si nejprve nahrajeme tyto knihovny, které obsahují chytré funkce využívané v celkovém programu.

```
#include "WiFi.h"
#include "esp_system.h"
#include "stdio.h"
#include <Arduino.h>
#include <ModbusIP_ESP8266.h>
#include <SPI.h>
#include <elapsedMillis.h>
#include <math.h>
```

Pro připojení k Wi-Fi je důležité znát SSID a heslo (password) pro připojení k síti, na kterou budeme zasílat informace.

```
const char *ssid = "SSID";
```

```
const char *password = "PASSWORD";
```

Dále jsme si vytvořili funkci, ve které bude probíhat veškeré připojování k síti a získávání IP adresy pro nadřazený systém. Připojování bude probíhat pomocí smyčky WHILE, dokud se nepřipojí na výše zadanou síť a jakmile se připojí, automaticky získáme vygenerovanou IP adresu.

```
void initWiFi() {
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    Serial.print("Connecting to WiFi ..");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print('.');
        delay(1000);
    }
    Serial.println(WiFi.localIP());
}
```

Musíme brát na vědomí, že se signál Wi-Fi může někdy odpojit, proto jsme vytvořili záchranou smyčku, která se při každém průchodu opakovací smyčky zeptá, jestli je Wi-Fi připojena. Pokud není, tak se vypojí a zpátky připojí na síť, jinak funguje dále beze změny. Pokud se stále nepřipojí, i přes toto zabezpečení, tak se o to znovu pokusí po 30 sekundách, pomocí časovače vytvořeného z funkce MILLIS().

```
void loop() {
    unsigned long currentMillis = millis();
    // if WiFi is down, try reconnecting every CHECK_WIFI_TIME seconds
    if ((WiFi.status() != WL_CONNECTED) && (currentMillis - previousMillis >=
interval)) {
        Serial.print(millis());
        Serial.println("Reconnecting to WiFi...");
        WiFi.disconnect();
        WiFi.reconnect();
        previousMillis = currentMillis;
    }
}
```

Po úspěšném připojení k síti získáme jednotlivé hodnoty z různých naprogramovaných funkcí. Tyto funkce získávají upravené signály z výše popsaných senzorů, které následně vypočítávají směr větru, průměrnou rychlost a rychlost okamžitou.

Pro směr větru jsme využili vnořené větvení pomocí IF a ELSE IF, kde se ptáme, jestli hodnota získána ze senzoru, je stejná jako rozmezí dané větve, pro kterou patří určitý směr. Hodnota je uložena do 16bitové proměnné u16WindDir.

```
void measureDir() {
    value = analogRead(36);
    if (value >= 50 && value <= 120) {
        u16WindDir = 1125;
    } else if (value >= 120 && value <= 164) {
        u16WindDir = 675;
    } else if (value >= 165 && value <= 220) {
        u16WindDir = 900;
    } else if (value >= 270 && value <= 300) {
        u16WindDir = 1575;
    } else if (value >= 440 && value <= 560) {
        u16WindDir = 1350;
    }
    .
    .
    .
}
```



```

} else if (value >= 2800 && value <= 3100) {
    ul6WindDir = 0;
} else if (value >= 3130 && value <= 3200) {
    ul6WindDir = 2925;
} else if (value >= 2690 && value <= 2993) {
    ul6WindDir = 3150;
} else if (value >= 2995 && value <= 3050) {
    ul6WindDir = 2700;
} else {
}
}

```

Pro měření rychlosti větru využíváme speciální časovač a čítač impulsů. Časovač pracuje na pozadí programu a jeho nastavení závisí na funkci timerAlarmWrite(). Musíme ho nastavit tak, abychom získali každou minutu přetečení při kterém se vykonání algoritmus. Nastavení je založené na taktovací frekvenci o hodnotě 80 MHz = 80 000 000 Hz. To znamená, že se bude inkrementovat 80 000 000 za sekundu. Pomocí funkce timerBegin() taktovací frekvenci před časujeme na 80 částí, tím pádem přetečení bude každou sekundu, a abychom získali přetečení každou minutu vložíme do funkce timerAlarmWrite hodnotu 60 000 000.

```

hw_timer_t *timer = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
timer = timerBegin(0, 80, true);
timerAttachInterrupt(timer, &onTimer, true);
timerAlarmWrite(timer, 60000000, true);
timerAlarmEnable(timer);

```

Výhoda tohoto časovače je, že nezastaví přerušením funkčnost programu, ale přerušením pracuje jenom s danou částí v paměti IRAM, kde proběhne část programu a čeká dále na další přetečení časovače.

Program funguje na jednom čítači, který se neustále inkrementuje, každým příchodem impulsu. My z něj získáváme okamžitou hodnotu a z ní odečtem předchozí hodnotu, abychom získali reálnou hodnotu za čas. Tuto hodnotu máme označenou jako VYSLEDEK, se kterým pracujeme a využíváme ho jak pro okamžitou rychlost, tak pro průměrnou.

Pro okamžitou rychlost využíváme rovnici, kde musíme zohlednit, že koeficient, který jsme si vypočetli na začátku je za sekundu. Zde musíme počítat za minutu, proto musíme do rovnice přidat časový převod.

$$v_s = 0,3 \text{ m/s} \quad (16)$$

$$v_{min} = \frac{0,3}{60} = 0,005 \text{ m/s} \quad (17)$$

Pro průměrnou rychlost sčítáme hodnoty okamžité do proměnné PRUMERNA pomocí podmínky IF. Jakmile bude proměnná *i* menší nebo rovna devět inkrementuje se, jinak se suma okamžitých hodnot vydělí 10 a zašle výsledek, aby mohla fungovat dál vynuluje proměnnou *i*.

```
pinMode(23, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(23), preruseni, RISING);void

preruseni() {
    count++;
}

void IRAM_ATTR onTimer() {
    portENTER_CRITICAL_ISR(&timerMux);
    vysledek = count - newcount;
    mzas = vysledek * 0.005F;
    if (i <= 9) {
        prumerna = prumerna + mzas;
        i++;
    } else {
        mzasprumerna = prumerna * 0.1F;
        i = 0;
    }
    newcount = count;
    portEXIT_CRITICAL_ISR(&timerMux);
}
```

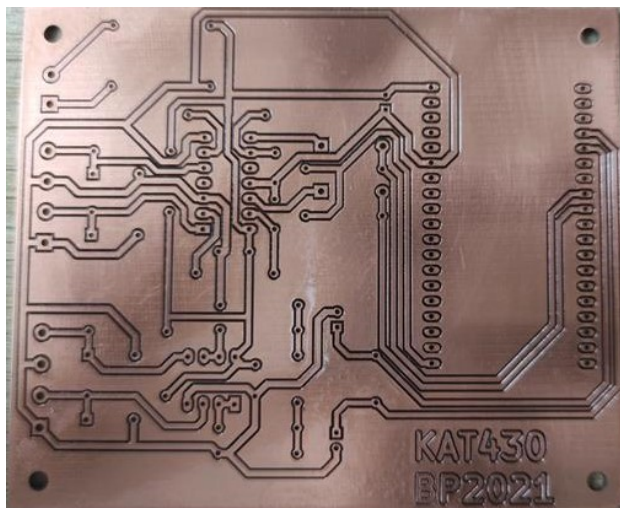
Zasílání výsledku na nadřazený systém využívá taky časovač na pozadí a tím taky neovlivní celkový program. Časovač je nastavený tak, že každou 1 ms inkrementuje a ukládá se do MEASURE, jakmile je MEASURE větší jak 10 sekund vynuluje se a zašle výsledky přes Wi-Fi na nadřazený systém.

```
elapsedMillis measure;
if (measure > 10000) {
    measure = 0;
    measureDir();
    int8_t rssi = WiFi.RSSI();
    Serial.print("RSSI: ");
    Serial.print(rssi);
    Serial.println(" dBm");
    mb.Ireg(100, u16Status);
    mb.Ireg(101, rssi);
    Serial.print("u16WindDir: ");
    Serial.print(u16WindDir*0.1F);
    Serial.println(" °");
    mb.Ireg(102, u16WindDir);
    u16WindSpeed = mzas * 100.0F;
    mb.Ireg(103, u16WindSpeed);
    u16WindSpeedAvg = mzasprumerna * 100.0F;
    mb.Ireg(104, u16WindSpeedAvg);
}
```

4 Realizace a ověření

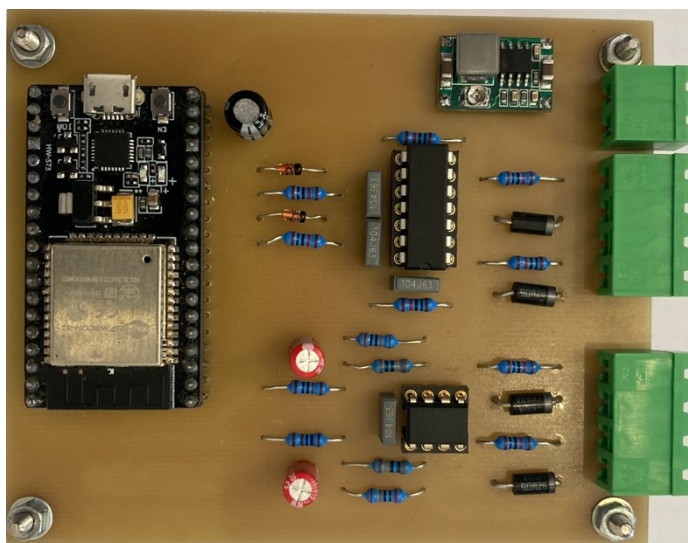
4.1 Realizace

Pro realizaci jsme využili na vytvoření DPS software KiCAD. V programu jsme si museli vytvořit některé součástky sami, jako je mikrokontrolér ESP32 nebo DC-DC měnič. Po dokončení návrhu plošného spoje jsme si nechali vyfrézovat plošky a vodivé cesty na cuprexitovou desku, která byla následně potřena ochranným lakem. Celá DPS je vytvořena z jedné spodní vrstvy.



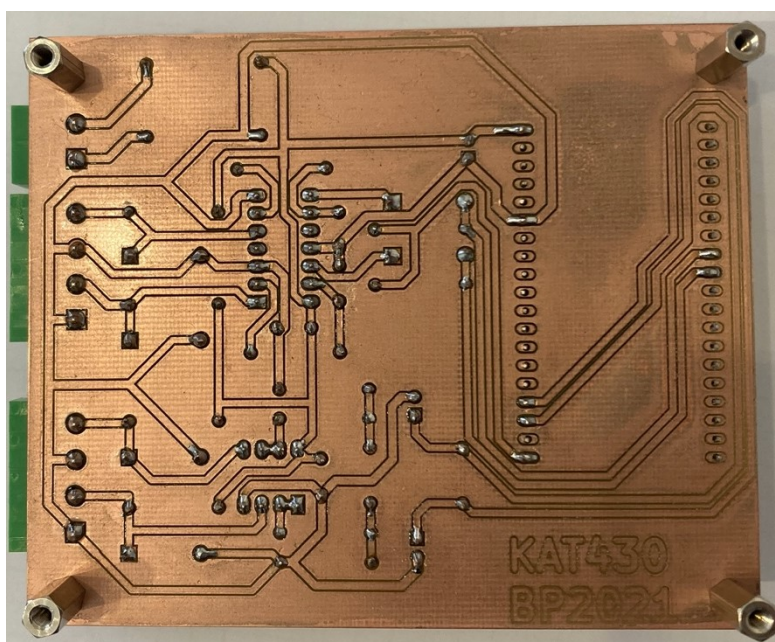
Obrázek 24: Vyfrézovaná deska plošných spojů na cuprexitu

Po vyfrézování jsme si desku osadili a následně zapájeli. Pro snadnější připojení ostatních komponentů, jsme využili výsuvné svorkovnice.



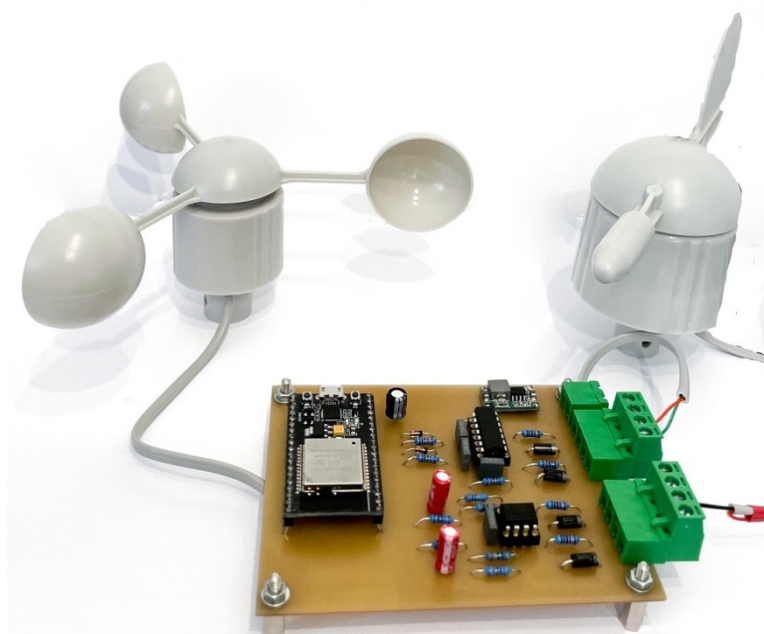
Obrázek 25: Osazená deska plošných spojů

Pro stabilitu desky jsme přidali distanční sloupky, které mohou být připevněny v konstrukci.



Obrázek 26: Zapájena deska plošných spojů

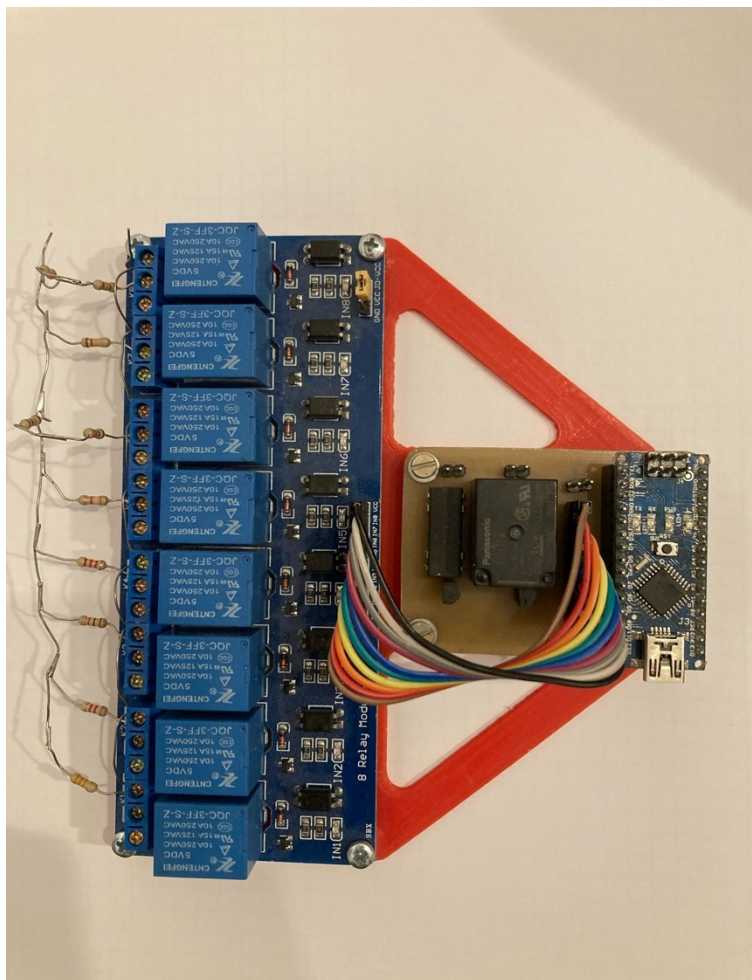
Pro uchycení jednotky a senzoru ještě není vytvořená konstrukce, ale zde ji vidíme jako plnohodnotnou jednotku pro měření směru a rychlosti větru.



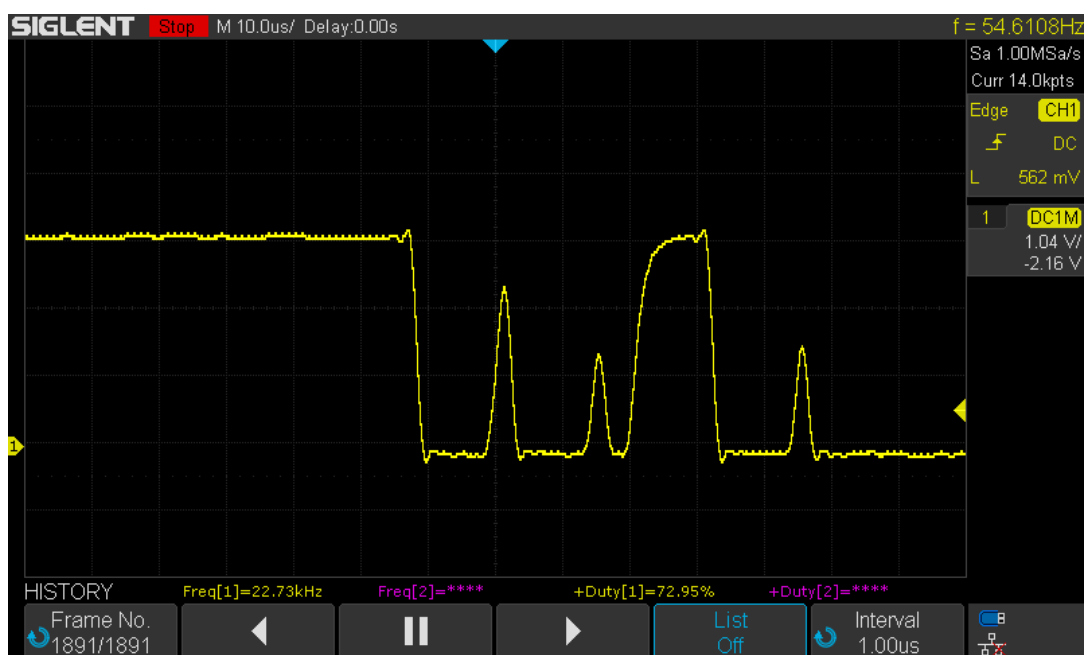
Obrázek 27: Soustava senzorů s jednotkou

4.2 Ověření

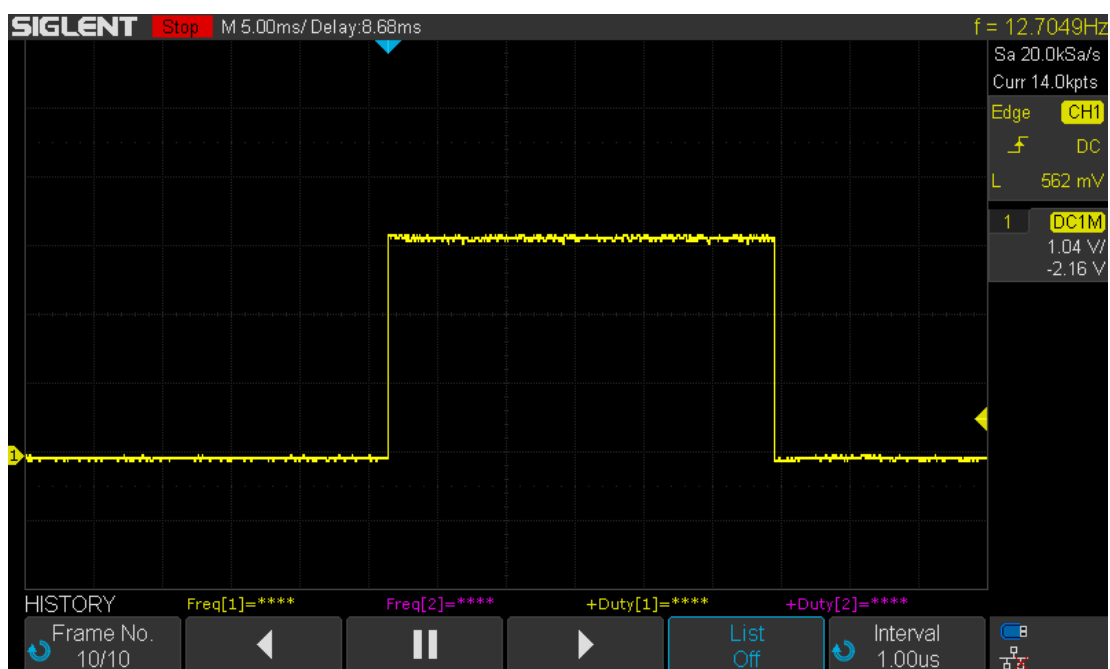
Pro ověření funkčnosti jsme využívali simulátor směru a rychlosti větru, který se dal regulovat pomocí příslušného softwaru. K vytvoření směru větru se využívalo spínání relé, které mělo pro každou světovou stranu jiný odpor. Na výstupní svorky jsme připojili napájecí napětí, díky kterých jsme později měřili úbytky a dále je vyhodnocovali. U vytvoření rychlosti větru byl spínač, který generoval impulzy podle rychlosti větru. Museli jsme také eliminovat zákmity spínacích zařízení a podle počtu impulzů z nich vyhodnotit rychlost větru.



Obrázek 28: Simulátor rychlosti větru a směru

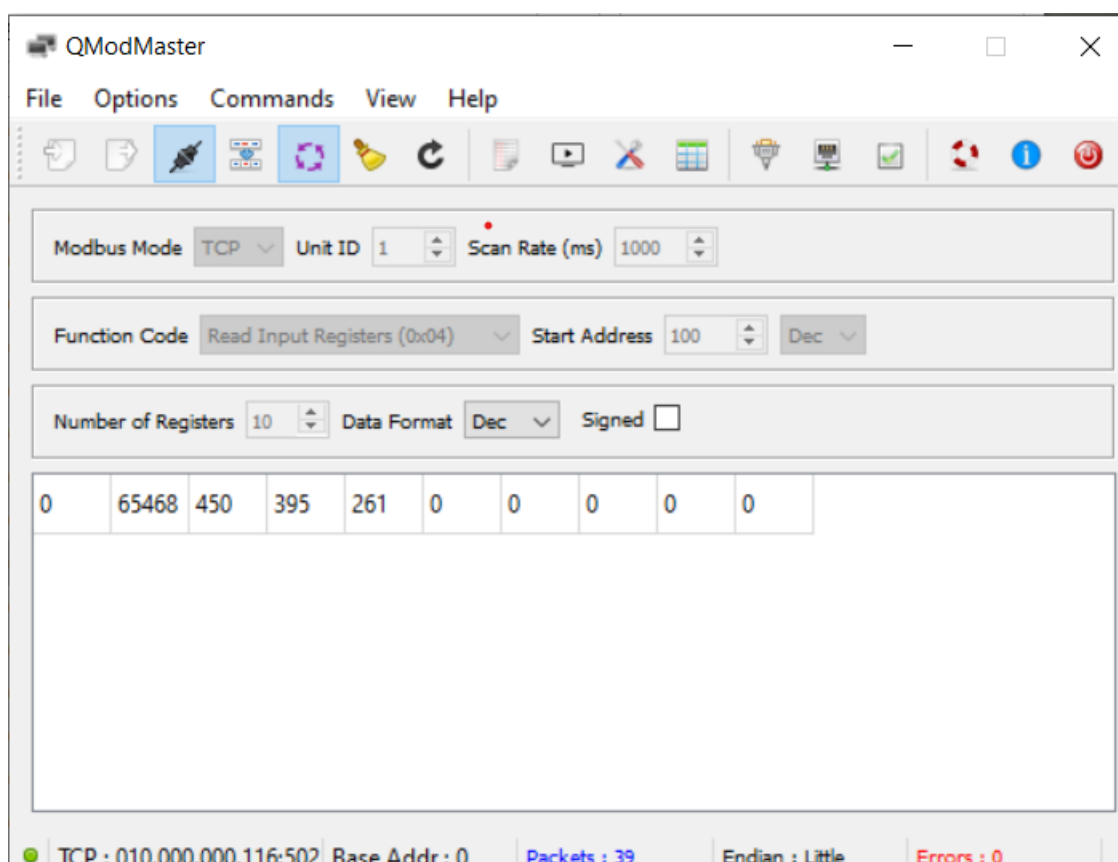


Obrázek 29: signál ovlivněný zakmitáváním



Obrázek 30: ošetřený signál proti zakmitávání

K ověření funkčnosti bezdrátového připojení jsme využili program QModMaster. Pro úspěšné připojení musíme nastavit IP adresu a TCP kanál, který je většinou 502. IP adresu získáváme od mikrokontroléru na začátku ze sériové komunikace.



Obrázek 31: Program QModMaster

Komunikaci převádíme do tabulkové podoby, která je rozdělena na jednotlivé měření. V druhém sloupci je síla připojení internetu, v třetím máme hodnotu rychlosti směru větru. Pro neztracení desetinného čísla jsme hodnotu v třetím sloupci vynásobili deseti. U čtvrtého a pátého sloupce je okamžitá rychlost větru a průměrná rychlost větru vynásobena stem.

Z obrázku 31 můžeme vyčíst, že se jedná o směr větru 45°, což odpovídá směru větru na severovýchodní světovou stranu. Okamžitá rychlost je 3,95 m/s a podle Beaufortovy stupnice je tato hodnota vnímána jako slabý vítr. U průměrné rychlosti je síla větru 2,61 m/s a vnímáme ji jako větřík.

Závěr

Cílem bakalářské práce bylo provést rešerši v metodách měření rychlosti větru a směru. Některé metody byly v práci shrnuty stručně, protože spočívají ve složitých fyzikálních jevech, ke kterým bych se musel více rozepsat. Nejsou zde uvedeny ani všechny, jelikož by se tomuto tématu dalo věnovat samostatně a zabralo by další celou bakalářskou práci. Objasnil jsem rozdíl mezi kontaktními a bezkontaktními metodami. V rešerši jsem se snažil hlavně určit výhody a nevýhody a také případy, ve kterých by toto měření mohlo být užitečné.

Hlavním cílem bylo navržení jednotky pro určení směru větru a rychlosti. Návrh jednotky jsem se snažil udělat bezporuchově a bezproblémově. Jednotka je připravena měřit rychlost větru a směru v normálních podmínkách a výsledky zasílá bezdrátovou formou. Můžu říct, že se jedná o stabilní systém.

Abychom jednotku mohli využít do extrémních podmínek museli bychom zde zohlednit závislost teploty na vlastnosti součástek, které na ně mají veliký vliv. Pro měření směru větru jsem tohle zohlednil a pro každý směr jsem nastavil v programu meze, které ochrání stabilitu systému. Použité metalické rezistory nejsou ideálními součástkami, které by držely stejnou hodnotu odporu. U měření větru jsem zjistil, co doopravdy je zakmitávání mechanických kontaktů a jaké můžou způsobit problémy. Tyto problémy jsem eliminoval Schmittovým spouštěčem a RC filtrem. Při návrhu jsem počítal s maximální naměřenou rychlosti větru na Fakultě elektrotechniky a informatiky. Jelikož se podnební podmínky mohou měnit, nastavil jsem filtr s rezervou, aby mohl přístroj fungovat v podmínkách s maximální rychlosti větru 171, 72 km/h.

Při realizaci jsem bohužel musel dělat na dvakrát desku plošných spojů. Tato chyba byla způsobena ESP32, kdy jsme zjistili až po osazení na desku, že část ADC nejde využít zároveň s bezdrátovým přenosem. Tuto chybu jsem po zjištění ihned opravil. Další problém se vyskytl při vybírání hodnot součástek, kde jsem využil pro měření směru větru RC filtru velký kondenzátor a velký rezistor o hodnotách $C=100\text{ uF}$ a $R=10\text{ k}\Omega$. Jakmile se začne větrná směrovka otáčet, mění tak úbytek napětí a pro velkou časovou konstantu se nám napětí mění pomalu, což by mohlo způsobit velkou chybu. Proto jsem změnil hodnoty RC filtru na $C=1\text{ uF}$ a $R=1\text{ k}\Omega$, touto změnou je měření dynamičtější.

Anemometr není vytvořen uživatelsky pro regulaci okamžité rychlosti a průměru rychlosti. Jsou zde nastaveny pevné časy, na kterých jsou vypočítány různé koeficienty, ale schopnější programátor by se ve zdrojovém kódu rozhodně vyznal a následně by si mohl upravit požadavky pro jednotku.

Jednotka není určena jenom pro tyto dva sensory, má čtyři vstupy, takže se dá různě modifikovat v podobě přidání dalších sensorů, na které budou působit stejné ochrany.

Při testování jsem dospěl k úspěšným hodnotám, test jsem prováděl pomocí zapůjčené simulace, díky které jsem se nemusel pokoušet o vlastní generování větru.

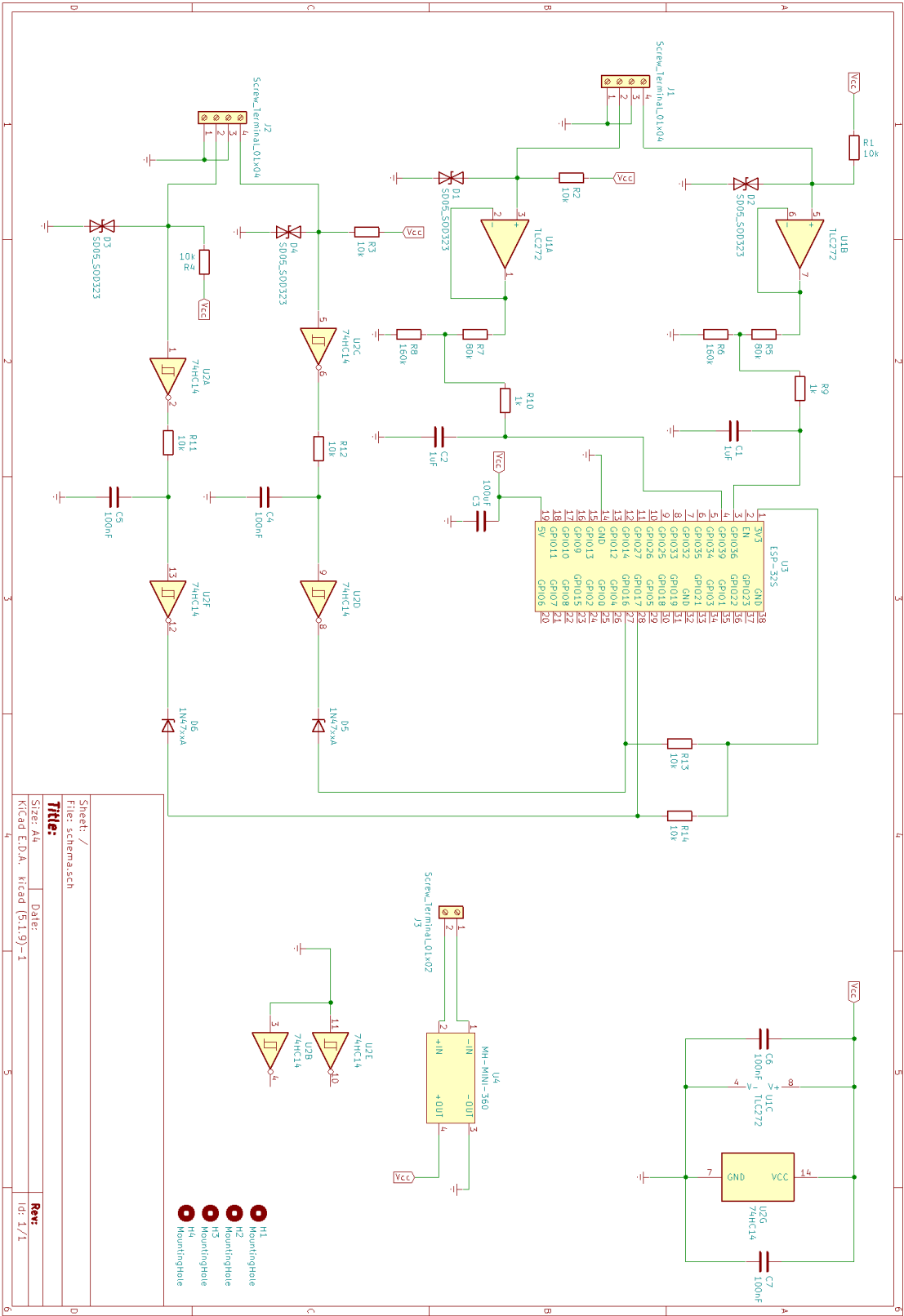
Literatura

- [1] National Geographic Society | National Geographic Society [online]. Copyright © 1996 [cit. 29.04.2021]. Dostupné z: <https://www.nationalgeographic.org/encyclopedia/anemometer/>
- [2] What Is the Coriolis Effect? | NOAA SciJinks – All About Weather. Home | NOAA SciJinks – All About Weather [online]. Dostupné z: <https://scijinks.gov/coriolis/>
- [3] Wikipedie: Otevřená encyklopedie: Beaufortova stupnice [online]. c2021 [citováno 28. 04. 2021]. Dostupný z WWW: <https://cs.wikipedia.org/w/index.php?title=Beaufortova_stupnice&oldid=19678224>
- [4] TECHNICKÁ UNIVERZITA V LIBERCI - PDF Free Download. Představujeme Vám pohodlné a bezplatné nástroje pro publikování a sdílení informací. [online]. Copyright © DocPlayer.cz [cit. 29.04.2021]. Dostupné z: <https://docplayer.cz/17354474-Technicka-univerzita-v-liberci.html>
- [5] Anemometr UNI-T UT363S - - Megaknihy.cz. MegaKnihy.cz - Levné Knihy Online - Megaknihy.cz [online]. Copyright © 2001 [cit. 29.04.2021]. Dostupné z: <https://www.megaknihy.cz/merici-pristroje/>
- [6] Skupina ČEZ [online]. Dostupné z: <https://www.cez.cz/edee/content/file/static/encyklopedie/pokusy/pokus14.html>
- [7] Domovské stránky uživatele [online]. Copyright © [cit. 29.04.2021]. Dostupné z: <http://home.zcu.cz/~formanek/mmvyuka/Data/ivk-mt-soubory/12-F.pdf>
- [8] [online]. Copyright © 6 [cit. 29.04.2021]. Dostupné z: https://www.mouser.com/catalog/additional/ATS_Qpedia_Dec07_Understanding%20hot%20wire%20anemometry9.pdf
- [9] Wind Vanes - What They Tell Us And How To Use Them - SwitchDoc Labs Blog. SwitchDoc Labs Blog - For Makers, by a Maker [online]. Copyright © 2021 [cit. 29.04.2021]. Dostupné z: <https://www.switchdoc.com/2018/04/wind-vanes-tell-us-use/>
- [10] Wikipedie: Otevřená encyklopedie: Světová strana [online]. c2021 [citováno 28. 04. 2021]. Dostupný z WWW: <https://cs.wikipedia.org/w/index.php?title=Sv%C4%9Btov%C3%A1_strana&oldid=19707956>
- [11] ESP32 Arduino: Timer interrupts - techtutorialsx. techtutorialsx - Lots of programming, IoT and technology tutorials [online]. Copyright © 2021 [cit. 29.04.2021]. Dostupné z: <https://techtutorialsx.com/2017/10/07/esp32-arduino-timer-interrupts/>
- [12] Anemometr WindSonic - ultrazvukový snímač větru | Automatizace.HW.cz. Automatizace.HW.cz | Elektronika v automatizaci [online]. Dostupné z: <https://automatizace.hw.cz/anemometr-windsonic-ultrazvukovy-snimac-vetru>
- [13] Vysoké učení technické v Brně [online]. Copyright © [cit. 29.04.2021]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=99920
- [14] Wikipedie: Otevřená encyklopedie: Modbus [online]. c2017 [citováno 28. 04. 2021]. Dostupný z WWW: <<https://cs.wikipedia.org/w/index.php?title=Modbus&oldid=15018113>>

Seznam příloh

Schéma zapojení.....	42
Netlist	43
Deska plošného spoje.....	44
Zdrojový kód.....	44

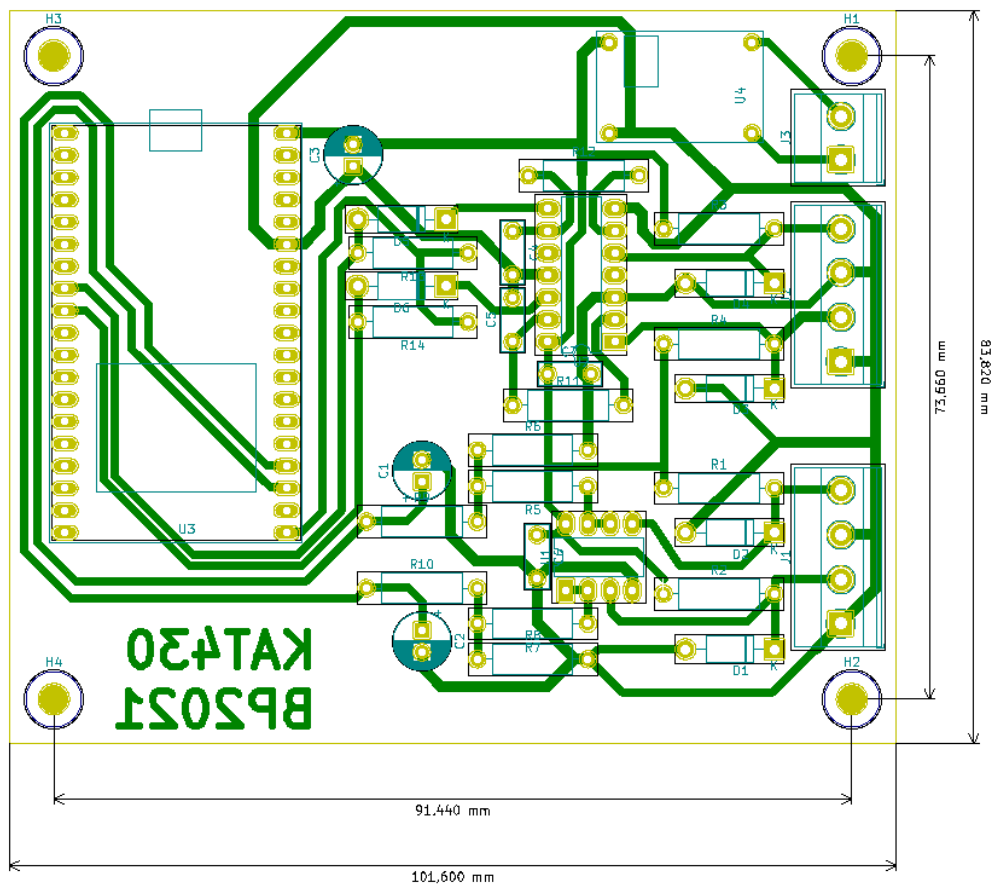
Schéma zapojení



Netlist

REF	Value	Part
U3	ESP-32S	ESP-32S-moje soucastky
U1	TLC272	TLC272
J1	Screw Terminal 01x04	Screw Terminal 01x04
R5	80k	Resistor
R6	160k	Resistor
R7	80k	Resistor
R8	160k	Resistor
R9	1k	Resistor
C1	1uF	Capacitor
R10	1k	Resistor
C2	1uF	Capacitor
J2	Screw Terminal 01x04	Screw Terminal 01x04
U2	74HC14	74HC14
R12	10k	Resistor
C4	100nF	Capacitor
R11	10k	Resistor
C5	100nF	Capacitor
D5	1N47xxA	1N47xxA
D6	1N47xxA	1N47xxA
R3	10k	Resistor
R4	10k	Resistor
R2	10k	Resistor
R1	10k	Resistor
U4	MH-MINI-360	MH-MINI-360-moje soucastky
J3	Screw Terminal 01x02	Screw Terminal 01x02
R13	10k	Resistor
R14	10k	Resistor
C3	100uF	Capacitor
D2	SD05 SOD323	SD05 SOD323
D1	SD05 SOD323	SD05 SOD323
D4	SD05 SOD323	SD05 SOD323
D3	SD05 SOD323	SD05 SOD323
C7	100nF	Capacitor
C6	100nF	Capacitor
H1	MountingHole	MountingHole
H2	MountingHole	MountingHole
H3	MountingHole	MountingHole
H4	MountingHole	MountingHole

Deska plošného spoje



Zdrojový kód

```
#include "WiFi.h"
#include "Wire.h"
#include "esp_system.h"
#include "stdio.h"
#include <Arduino.h>
#include <ModbusIP_ESP8266.h>
#include <SPI.h>
#include <elapsedMillis.h>
#include <math.h>

const char *ssid = "TESTMS";
const char *password = "430autobus2020";

ModbusIP mb;
uint16_t u16Status = 0x0000;
unsigned long previousMillis = 0;
unsigned long interval = 30000;
elapsedMillis measure;

uint32_t count;
uint32_t newcount = 0;
uint32_t i = 0;
float mzas, mzasprumerna, prumerna;
uint16_t value;
uint32_t vysledek;
```

```

uint16_t ul6WindSpeed;      //x100
uint16_t ul6WindSpeedAvg;  //x100
uint16_t ul6WindDir;       //x10

hw_timer_t *timer = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;

void preruseni();
void IRAM_ATTR onTimer();
void measureDir();
void initWiFi();

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    timer = timerBegin(0, 80, true);
    timerAttachInterrupt(timer, &onTimer, true);
    timerAlarmWrite(timer, 6000000, true);
    timerAlarmEnable(timer);
    pinMode(16, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(16), preruseni, RISING);
    initWiFi();
    Serial.print("RSSI: ");
    Serial.println(WiFi.RSSI());
    mb.server();
    mb.addIreg(100, 0, 10);
    measure = 10000;
}

void loop() {
    unsigned long currentMillis = millis();
    // if WiFi is down, try reconnecting every CHECK_WIFI_TIME seconds
    if ((WiFi.status() != WL_CONNECTED) && (currentMillis - previousMillis >=
interval)) {
        Serial.print(millis());
        Serial.println("Reconnecting to WiFi...");
        WiFi.disconnect();
        WiFi.reconnect();
        previousMillis = currentMillis;
    }
    mb.task();
    if (measure > 10000) {
        measure = 0;
        measureDir();
        int8_t rssi = WiFi.RSSI();
        Serial.print("RSSI: ");
        Serial.print(rssi);
        Serial.println(" dBm");
        mb.Ireg(100, ul6Status);
        mb.Ireg(101, rssi);
        Serial.print("ul6WindDir: ");
        Serial.print(ul6WindDir*0.1F);
        Serial.println(" °");
        mb.Ireg(102, ul6WindDir);
        ul6WindSpeed = mzas * 100.0F;
        mb.Ireg(103, ul6WindSpeed);
        ul6WindSpeedAvg = mzasprumerna * 100.0F;
        mb.Ireg(104, ul6WindSpeedAvg);
    }
}

```

```

void measureDir() {
    value = analogRead(36);
    if (value >= 50 && value <= 120) {
        ul6WindDir = 1125;
    } else if (value >= 120 && value <= 164) {
        ul6WindDir = 675;
    } else if (value >= 165 && value <= 220) {
        ul6WindDir = 900;
    } else if (value >= 270 && value <= 300) {
        ul6WindDir = 1575;
    } else if (value >= 440 && value <= 560) {
        ul6WindDir = 1350;
    } else if (value >= 680 && value <= 800) {
        ul6WindDir = 2025;
    } else if (value >= 880 && value <= 970) {
        ul6WindDir = 1800;
    } else if (value >= 2100 && value <= 2250) {
        ul6WindDir = 225;
    } else if (value >= 1500 && value <= 1650) {
        ul6WindDir = 450;
    } else if (value >= 2930 && value <= 2970) {
        ul6WindDir = 2475;
    } else if (value >= 2980 && value <= 3050) {
        ul6WindDir = 2700;
    } else if (value >= 2400 && value <= 2500) {
        ul6WindDir = 3375;
    } else if (value >= 2600 && value <= 2830) {
        ul6WindDir = 0;
    } else if (value >= 2850 && value <= 2950) {
        ul6WindDir = 2925;
    } else if (value >= 2960 && value <= 2993) {
        ul6WindDir = 3150;
    } else if (value >= 2190 && value <= 2250) {
        ul6WindDir = 2250;
    } else {
        /*Serial.println("chyba");*/
    }
}

void preruseni() {
    count++;
}

void IRAM_ATTR onTimer() {
    portENTER_CRITICAL_ISR(&timerMux);
    vysledek = count - newcount;
    mzas = vysledek * 0.005F;
    if (i <= 9) {
        prumerna = prumerna + mzas;
        i++;
    } else {
        mzasprumerna = prumerna * 0.1F;
        i = 0;
    }
    newcount = count;
    portEXIT_CRITICAL_ISR(&timerMux);
}

void initWiFi() {

```

```
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi ..");
while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
}
Serial.println(WiFi.localIP());
}
```